WHAT IS CLAIMED IS:

1. A method of constructing a hardware architecture for transform functions, comprising the steps of:

a setting-up step of a transform function, to select a transform function which transfers an input signal x(n) on a domain into an output signal y(k) on another domain;

a simplifying step of value-specific transform coefficients, to simplify each group of transform coefficients with the same value as an identical transform coefficient, wherein every identical transform coefficient is respectively processed by a fixed-one-input multiplier;

a multiplying step, to separately use the fixed-one-input multipliers for multiplying the input signals by the value-specific transform coefficients and generating the intermediate results;

a distributing step, to use a path-selector to distribute the product results to accumulators according to the timing diagrams of the output signalse;

an accumulating step, to use the accumulators to perform the accumulations at the correct timing diagrams to generate the accumulated results;

a constant multiplying step, to use the multipliers to multiply the accumulated results by a constant-value item of the transform function and generate the output signals; and

an outputting step, to output the output signals.

2. The method as claimed in claim 1, wherein the transform function

is $y(k) = A\sum_{n=0}^{N-1} T_c(k,n)x(n)$ for $k = 0,1,2,\cdots,N-1$, where A is the constant item and $T_c(k,n)$ is the corresponding transform coefficient.

3. The method as claimed in claim 2, wherein the transform function is applied to perform an inverse discrete Fourier transform (IDFT) for $A=\frac{1}{N}$.

4. The method as claimed in claim 1, further comprising a simplifying step of symmetry-based transform coefficients after the simplifying step of transform coefficients to simplify symmetric transform coefficients for sharing a fixed-one-input multiplier.

5. The method as claimed in claim 1, wherein the transform coefficients are represented in a binary form.

6. The method as claimed in claim 5, wherein each of the fixed-one-input multipliers respectively computes the corresponding transform coefficient consists of at least one addition or subtraction unit.

7. The method as claimed in claim 6, wherein the multiplying step comprises the steps of:

determining values of all transform coefficients;

analyzing the bit values of transform coefficients for extracting shared items, wherein each shared item is calculated by the addition and/or subtraction units; and

trying to construct the values of transform coefficients by using the shared items.

8. The method as claimed in claim 7, wherein the transform coefficients are represented by a canonic signed digit (CSD).

9. The method as claimed in claim 7, wherein the transform coefficients are represented by a hybrid signed digit (HSD).

10. An apparatus of constructing a hardware architecture for transform functions, comprising:

an input unit to receive an input signal and then distribute the input signal to at least one fixed-one-input multiplier;

at least one fixed-one-input multiplier to multiply the input signal with the transform coefficients defined in the transform function and generate product results;

at least one path-selector to distribute the product results to accumulators according to the timing diagrams of the output signals based on the definition of the transform function;

at least one accumulator to correspond to at least one timing diagram of the output signals and accordingly receive the product results for accumulation to generate accumulated results; and

an output unit to output the output signals.

11. The apparatus as claimed in claim 10 further includes at least one multiplier to multiply the accumulated results by a constant value of the transform function in order to calculate the output signals.

12. The apparatus as claimed in claim 10, wherein the transform function is $y(k) = A\sum_{n=0}^{N-1} T_c(k,n)x(n)$ for $k = 0,1,2,\cdots,N-1$, where A is the

constant item and $T_c(k,n)$ is the corresponding transform coefficient.

13. The apparatus as claimed in claim 10, wherein the transform coefficients are represented in a binary form.

14. The apparatus as claimed in claim 13, wherein each of the fixed-one-input multipliers respectively computing the corresponding transform coefficient consists of at least one addition and/or subtraction unit.

15. The apparatus as claimed in claim 10, wherein the path-selector further comprises a controller to generate the control signals.